Micro-Mock

Table of Contents

A micro Kotlin/Multiplatform Kotlin Symbol Processor that generates Mocks & Fakes.

Limitations:

- Mocking only applies to **interfaces**
- Faking only applies to **concrete tree**

Usage

Mocks

> **CAUTION** Only **interfaces** can be mocked!

Requesting generation

You can declare that a class needs a specific mocked interface by using the `@UsesMocks` annotation.

```
@UsesMocks(Database::class, API::class)
class MyTests {
}
```

Once a type appears in `@UsesMocks`, the processor will generate a mock class for it.

Defining behaviour

To manipulate a mocked type, you need a `Mocker`. You can then create mocked types and define their behaviour:

```
@UsesMocks(Database::class, API::class)
class MyTests {
    @Test fun myUnitTest() {
        val mocker = Mocker()
        val db = MockDatabase(mocker)
        val api = MockAPI(mocker)

        mocker.on { db.open(isAny()) } returns Unit ①
        mocker.on { uv.getCurrentUser() } runs { forgeFakeUser() } ②
    }
}
```